

Breaking out that Perl script: The imaging and imagining of code in The Social Network and Catfish

Dinnen, Zara

DOI:

[10.1386/ejac.32.2.173_1](https://doi.org/10.1386/ejac.32.2.173_1)

License:

Other (please specify with Rights Statement)

Document Version

Early version, also known as pre-print

Citation for published version (Harvard):

Dinnen, Z 2013, 'Breaking out that Perl script: The imaging and imagining of code in The Social Network and Catfish', *European Journal of American Culture*, vol. 32, no. 2, pp. 173-186.

https://doi.org/10.1386/ejac.32.2.173_1

[Link to publication on Research at Birmingham portal](#)

Publisher Rights Statement:

This is the pre-publication version of the article published in *European Journal of American Culture*, Vol 32, No 2 (2013)
http://dx.doi.org/10.1386/ejac.32.2.173_1.

Checked July 2015

General rights

Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

- Users may freely distribute the URL that is used to identify this publication.
- Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
- User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
- Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy

While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

Final article published as: Dinnen, Z. (2013), 'Breaking out that Perl script: The imaging and imagining of code in *The Social Network* and *Catfish*', *European Journal of American Culture*, 32: 2, pp. 173-186, doi: 10.1386/ejac.32.2.173_1

This is the pre-publication version. Final article published as:

Dinnen, Z. (2013), 'Breaking out that Perl script: The imaging and imagining of code in *The Social Network* and *Catfish*', *European Journal of American Culture*, 32: 2, pp. 173-186, doi: 10.1386/ejac.32.2.173_1

Zara Dinnen

"Breaking out that Perl script": The imaging and imagining of code in *The Social Network* and *Catfish*

Zara Dinnen

Abstract

For most users of digital technology, code is the hidden engine of their experience: silent, disguised, unknown and possibly unknowable. It is this experience of code that will be the subject of this article. Code, which is often perceived as textual, runs software, which is predominantly received as graphical. As with printing and handwritten script, all text is technically graphical, but this observation refers specifically to the increasing use of graphic icons, and visual media, to form the language of digital communications – the egg-timer, the pointer, Twitter and Facebook 'buttons', video CVs and avatars

This article will consider how the user encounter with code, as image, can perhaps be brought to our attention through visual narratives that attempt to represent digital communication and programming. The article will begin with a consideration of code itself as a problematic imagetext: that is, code as a semantic, or linguistic, material that is not readable in a human context; that persists as a sign until the instance of its being run – an instance of potential performativity. It will do so through a discussion of critical interventions in the field of code studies by scholars such as Wendy Hui Kyong Chun, Alexander Galloway and Adrian Mackenzie. Having established this reading, the article will move on to consider how code might resist representation in narrative form, particularly in film. It will do so through a discussion of Matthew Kirschenbaum's term 'medial ideology', a construct that references fictional representations of code and coding that wilfully obscure the actual mechanical process. The main section of the article will be given over to tracing Kirschenbaum's term through two recent films that creatively interpret the difficulties of depicting computing and communication on screen, *The Social Network* (Fincher, 2010), and *Catfish* (Joost and Schulman, 2010).

Keywords

digital culture

code studies

Matthew G. Kirschenbaum

The Social Network

Catfish

Introduction: Matters of code

For most users of digital technology, code is the hidden engine of their experience: silent, disguised, unknown and possibly unknowable. Code as the formative material and symbolic operation of digital culture remains difficult to comprehend within everyday computer use, mostly encountered unwittingly in the form of glitches in software. This article will begin from the premise that code is a complex imagetext, instantiated at the level of the user–computer encounter, through a graphical user interface (GUI). Source code that is perceptibly textual is only operational when it is run as object code within a computational, not human-readable, context.¹ The performativity of code does not manifest itself as part of the GUI; the moment a user encounters code it is not as (or in) action, but it is suppressed within a system of (human-readable) signs and images – the egg-timer, the pointer, Twitter and Facebook 'buttons', Skype phones and Gmail envelopes. This article will consider how the user encounter with code, as image, can perhaps be brought to our attention through visual narratives that attempt to represent digital communication and programming. It will question how cinematic narratives that depict code as a process of programming or hacking within the framework of a realist, or 'everyday', plot and style might articulate the procedural matter of code. The films discussed here point to the way code seems to resist narrative; this resistance can be positioned as a productive and generative site for thinking about digital culture and our capacity to represent it as story.

In the essay 'Word as Image', Matthew G. Kirschenbaum articulates the difference between word and image in digital representation. Kirschenbaum (2003: 138–39) notes, 'as computational data structures, images differ radically and fundamentally from electronic text'; '... images remain largely opaque to the algorithmic eyes of the machine'. Kirschenbaum's argument articulates the structural difference between the ways that code manifests image and word. The processes Kirschenbaum describes refer to the back-end work of digital computing. Contemporary computation oversees a development of the blurring of pictorial and alphanumeric languages, which commences with the computer itself as a medium. As Kirschenbaum (2003: 141) argues, '... words and images ... remained technically distinct from one another throughout every prepress process up until the rise of desktop publishing in the 1980s'. This article will suggest that the difficulty of representing code in, and as, narrative is elucidated by thinking of code as imagetext. Doing so enables an approach to code that takes into account its shifting manifestation as language, sign and performative language, or procedure. In addition, it frames the depiction of code on screen – in scenes of programming – as simultaneously image and text.

This article will begin with a consideration of code itself as a problematic imagetext: that is, code as a semantic, or linguistic, material that is not readable in a human context; that persists as a sign until the instance of its being run – an instance of potential performativity. Having established this reading, the article will move on to consider how code might resist representation in narrative form, particularly in film. It will do so through a discussion of Kirschenbaum's (2008) term 'medial ideology', and of two

recent films that creatively interpret the difficulties of depicting computing and communication on screen, *The Social Network* (Fincher, 2010), and *Catfish* (Joost and Schulman, 2010).

Code as imagetext

From emoticons in e-mails to multimedia flash productions, digital culture is marked by its application of images, encouraging users to speak – to make their mark – through a pictorial language. Sherry Turkle views this kind of language use in terms of its immediacy and moderation as something between text and speech. Turkle (1997: 183) writes that 'this new writing is a kind of hybrid: speech momentarily frozen into artefact, but curiously ephemeral artefact ... emoticons replace physical gestures and facial expressions ... the new writing is somewhere in between traditional written and oral communication'. Turkle is here referring to the text-based virtual environments that populated computer networks in the mid-1990s. Whilst these multi-user domains/dungeons (MUDs) have been replaced by the sophisticated graphical imagery of massive multi-player online games, and virtual environments such as 'Second Life', the pervasive use of text-based communications through SMS, various message apps for smartphones and closed messenger services such as BlackBerry's BBM suggests the persistent presence of this 'hybrid' language. This language is a form of communication that elides the formality of production in the digital medium with a colloquial iterability performed through speech – what Caroline Bassett has called 'speaking-writing' and 'seeing-saying' (Bassett 2007: 129).

The oscillation between text and image that is practised in digital communication is, perhaps, a manifestation of the coming together of text and image as interface. Digital devices rely on image-based interfaces to orientate users unfamiliar with programming code. The very nature of the GUI is one of effacement. It is there to represent as 'directly' as possible a tool, or object, that is in reality a complex coordination of processes. Describing the prevalence of the screen as a metaphor for digital culture more generally, Marianne van den Boomen (2009: 254) suggests that the function of the screen as the sticking point for 'knowing' the digital, code, 'may be explained by the ability of the GUI to translate and articulate all other components into visual representations, thus rendering irrelevant what remains invisible, and rendering unthinkable what remains unmetamorphised'.² As touch screens proliferate in computing culture, the active effacement performed by the GUI is amplified. Whilst the GUI is put in place to suppress code – to give it a navigable public face – it is also the thing that substantiates code. It is through the interface that code materializes, that a user commands code. In a user's experience of code, the image, that is, the interface, is the thing that both conceals and manifests code.

Throughout discourses of New Media, the Digital Humanities, and Science and Technology Studies, recent scholarship has called for an intervention in the critical appreciation of code. Writing in the last decade by Wendy Hui Kyong Chun, Johanna Drucker, Alexander R. Galloway, Katherine Hayles, Matthew G. Kirschenbaum, Adrian Mackenzie and Eugene Thacker, to name some, has been concerned with the study of the materiality of code, of its textual properties and of its appearance at an aesthetic

and cultural level. These writings intersect and respond to each other in different ways, but across the discourse as a whole there is a shared concern with code as language and code as text. In *Protocol*, Alexander R. Galloway describes his own treatment of code as textual as a position permitted by the multiplicity of the term textual, rather than the literary-linguistic qualities of code. Galloway (2004: xxiv) writes:

Like film was to André Bazin or fashion was to Roland Barthes, I consider computers to be a fundamentally textual medium. The reason is obvious: computers are based on a technical language called code. This underlying code provides the textual link between computers and critical theory.

In referencing Barthes and Bazin, Galloway finds his 'textual link between computers and critical theory', and yet vast portions of *Protocol* argue against code as something that can be textually appreciated. Code is repeatedly asserted as, by definition, '*against interpretation*' (Galloway 2004: 52, emphasis in original). By reading code as against interpretation, as a site of resistance, Galloway presupposes a critical binary – he is unable to think of code outside the terms of interpretation, even as he posits code against those terms. This paradoxical turn – to think of code against the critical theory used to describe it – is itself the site of Galloway's encounter, and of the critical and cultural work that responds to questions of computing code.

In *Cutting Code* (2006), Adrian Mackenzie performs a series of close readings of code in application. It is possible to consider *Cutting Code* as positioned at exactly

the point at which code resists such interpretive narrativizing. There is a critical point at which code does and does not elude our attempts to apprehend it. As Mackenzie (2006: 2) writes:

Borrowing a concept from physics, we could say that software undergoes phase transitions, or changes of state. It solidifies at some points, but vaporizes at others ... The characteristics of software as a material object, as a means of production, as a human-technical hybrid, as medium of communication, as terrain of political-economic contestation – in short as sociality – seem hard to represent.

Mackenzie's assertion reflects Wendy Hui Kyong Chun's proposition that code functions in a way akin to law: that is, code not as law, or like law, but as the ultimate law that society cannot implement, only aspire to:

... software is code, that code is – has been made to be – executable, ... this executability makes code not law but rather every lawyer's dream of what law should be: automatically enabling and disabling certain actions and functioning at the level of everyday practice.

(Chun 2011b: 101)

Code in the digital context is another, an unknowable material whose agency we approach through our own complex critical language. This is a position that Chun

maintains through her insistence on understanding coding procedures as distinct stages: source code, and object code – that is code as it is run through a compiler (in a computational context). Whilst source code is produced by human programmers, for Chun the action of code is the instance of its being run (as object code); in this action, this performance, code is not human-readable. Chun asserts that thinking of source code as 'code' more generally can only happen after the fact, after its having been compiled and executed. Through this breakdown of the performative qualities of code, Chun establishes a sense of code as other – operating in a computational context that is not open to critical interpretative work.³

In the essay 'Codes and Codings in Crisis', Mackenzie and co-author Theo Vurdubakis explicitly describe this state. Mackenzie and Vurdubakis use Jacques Derrida's and Judith Butler's theories of performativity as a way of apprehending code – or rather of apprehending its misapprehension. They argue that

any account of coding that starts or ends by opposing code to language, by separating meaning and force, or communicating and acting, is bound for trouble ... the iterability on which performativity relies – the repetition that allows it to become conventional – is itself coded.

(Mackenzie and Vurdubakis 2011: 7)

Mackenzie and Vurdubakis describe the ways that code is knowable through language. They suggest that the relationship between code and critical theory (theory based on linguistic descriptions of code) is formed by codes continually deferring to

other codes. In this way, as with Chun's 'law' and Galloway's 'against interpretation', Mackenzie and Vurdubakis assert that code resists exact representation in language: it is always problematic. Both programming code and language are understood to instantiate complex relations between iterability and executability. Rather than attempt to unpick these relations, the approach in this article is to keep hold of the theoretical knots that thinking about code involves. This is particularly necessary to consider how code can be critically rendered in visual culture, and to measure how critical that culture is able to be about code.

Code at the multiplex

The critical work discussed in the previous section, particularly that by Chun and Mackenzie, is primarily concerned with the way code operates in everyday spaces (see also Kitchin and Dodge 2011). This work is formative to my approach, and must be read in dialogue with work in the field of social studies, which insists on considering the Internet in the context of everyday use (see Bakardjieva 2005; Haythornwaite and Wellman 2002). That said, this article is not focused on the experience of code in the everyday, but the way in which recent films might represent, or produce, an awareness of code in the everyday. Pursuing this line of enquiry is to think about cinematic (and as part of a broader scope, literary) representations of digital culture that posit the material and symbolic processes of code against the digital sublime as it has been iterated in science fiction and cyberpunk. This approach is to follow that of Kirschenbaum, who, in *New Media and the Forensic Imagination* (2008), highlights the ways a materialist reading of computers must necessarily counter overriding cultural assumptions about

digital media. He describes such assumptions as 'medial', arguing that cultural representations of computer use (such as film and literature) tend to science-fictionalize the actual processes possible with digital media. For Kirschenbaum (2008: 36) this contributes to a 'medial ideology'. A medial ideology eschews representations of a 'comprehensive treatment of the material particulars of a given technology' in favour of 'popular representations of a medium, socially constructed and culturally activated to perform specific kinds of work' (Kirschenbaum 2008: 36). Kirschenbaum (2008: 38–40) cites the cyberspaces of the novel *Neuromancer* (Gibson 1984) and the film *TRON* (Lisberger, 1982) as being typical 'artificial alloys' that stand in place of 'inscription, mechanism, sweat of the brow ..., and cramp of the hand'.

In a review of Kirschenbaum's book, Johanna Drucker (2009) describes 'medial ideology' as 'the seductive tendency to read "the digital event on the screen" as if it existed independent of the specific technological mechanisms on which it depends'. In the review, Drucker suggests that this 'seductive tendency' is due to the privileging of the screen that already pervades culture and cultural studies, that the fantastical representation of the digital is always already more appealing than a mechanical one (Drucker 2009). This article will not offer a new formal approach to the films it reads (as Kirschenbaum does) – that is the work of a longer project. Instead, it will focus on the way in which what is seen on screen might point a viewer towards what is not seen: the coded material of digital culture. It will follow Kirschenbaum in using his description of 'artificial alloys' – in the case of programming this might be hacker culture (as it is referenced by the shorthand of fashion and music), the speed of inputting source code

and the rapid compiling of object code (the latter not shown, although often visualized as a simulation) – to think about how cinema might produce alternate models to frame the task of representing code in its material and symbolic state. By focusing on films that themselves offer up a realist style – connected to the everyday – this article is therefore not engaged with reading such seminal 1990s' depictions of hacking and coding as appear in *Ghost in the Shell* (Oshii, 1995) and *The Matrix* (Wachowski and Wachowski, 1999), and popular films *The Net* (Winkler, 1995) and *Hackers* (Softley, 1995).⁴ Instead, it will focus on two recent (2010) cinematic representations of Facebook in its everyday use and at its moment of conception. In doing so, it will consider the way in which realist modes are used to draw attention to the otherness of code and software, whilst emphasizing this otherness as itself a material concern of digital culture.

The assertion made in this article is that medial ideologies pervade representations of digital technology in popular culture (come to stand in place of analysis of the digital material itself), because it is, to an extent, only through the imaginary that we can access what happens on the other side of the screen. This is to suggest that if code is not a human language – and in fact we perform a kind of textual studies with code in order to find a way of somehow rendering it understandable, if not readable – then perhaps code (as the mechanical driver of computer software) actually resists narrative. That there is such a strong medial ideology surrounding digital technology might not be a symptom of people's reluctance to fully understand that technology; rather, it might indicate the impossibility of that task. Clearly, computers are

man-made machines, and are of course knowable, mechanically. But the tendency to anthropomorphize, or mystify, the running processes of computing suggests a cultural need to narrativize these processes beyond the formal, materialist approaches undertaken by Kirschenbaum.⁵ Instead, representations of computing and coding rely on extrapolation to depict that which is not visible to the naked eye, or readable through the structures of human language. The next section of this article will develop this line of enquiry by considering two recent films that seem to play with Kirschenbaum's term 'medial ideology': *The Social Network* and *Catfish*. Both films take as their main subject the social networking site Facebook. In *Catfish*, the subject of the movie is the way we encounter and use Facebook in a formative capacity in our everyday lives. *The Social Network* tells the story of Facebook's conception, and of its founder, Mark Zuckerberg. The discussion of the films will focus specifically on how each tries to narrativize digital media, how they attempt to visually represent the digital in a way that can be recognized and assimilated within a cinematic format.

Code on film (*The Social Network*)

The Social Network depicts multiple scenes of code-writing, particularly hacking, mostly performed by Harvard undergraduates – depicted as geeks in Gap hoodies, drinking light beer and doing shots. In a review of the film for the *New York Review of Books*, Zadie Smith (2010: n.p.) wrote of the dilemma director David Fincher must have had when deciding whether to include these scenes:

how to convey the pleasure of programming ... in a way that is both cinematic and comprehensible? Movies are notoriously bad at showing the pleasures and rigors of art-

making, even when the medium is familiar ... Programming is a whole new kind of problem.

One of the first sequences of the movie shows the character of Mark Zuckerberg (real-life founder of Facebook, played in the film by actor Jesse Eisenberg) arriving home after being dumped by his girlfriend, grabbing a bottle of beer and sitting down to write a blog post about the girl he was just dumped by. This scene takes place at a desktop computer but the character also has a laptop next to the desktop screen. Open on the laptop is a Harvard website that shows student profiles. Zuckerberg's roommate jokingly suggests comparing girls' profiles to pictures of animals and rating them. Zuckerberg decides that rather than compare girls' profiles to animals, he should set up a program to rate girls' photos against each other. This thought process is shown on screen as Zuckerberg types it all in his blog and it is simultaneously narrated by a voice-over (Eisenberg playing Zuckerberg reading the blog aloud). After Zuckerberg has hit on this idea, a quick-cut sequence of programming and blogging commences with his call: 'Let the hacking begin'. In the scene that follows, the voice-over continues to narrate the blog post that describes the hacking sequence, while on screen the film alternates between images of the two screens (the blog post and the code for hacking into the Harvard student profile sites). The whole scene is cut up with action taking place on campus elsewhere: girls partying with (but mostly for) guys in the exclusive clubs Zuckerberg covets. The overall effect of the scene conflates the writing of the code with the actions it will eventually affect. Through montage and compressed

temporal gestures, Zuckerberg is shown to be almost literally rewriting the hierarchical nature of social networking (as it is represented by Ivy League institutions) by the act of programming. In this sequence, iterability and executability, through the logic of performativity, are blurred by Fincher's edits.

Alongside the contrasting depictions of social networking played out in this scene are the contrasting practices of reading and writing represented on screen as the blog post/voice-over and programming code. In the scene, the depictions of writing flicker between writing plain text and HTML (for the blog) and writing command code (to build the rating site 'facemash'). These are two different languages. Both are code but HTML is written in plain text (natural language) with additional rules that are deployed to dictate the appearance of that text online. The code for 'facemash' is a structural layer below this (such as C++) and falls into the non-textual category of code discussed so far in this article. The ellipses between Zuckerberg's blog text (readable) and his program text (code) serve to give an impression of fluidity to the procedures for writing structural code. It is made to seem almost immaterial, as the program is executed and produced almost simultaneously. In this scene, the voice speaks over a pounding techno score (by Trent Reznor), and this one long scene comprises many short, quick edits: the instant 'live-ness' of facemash is depicted as boys in other dorms get sent a link to the site and begin rating girls, and this feeds back to the parties that Zuckerberg had failed to infiltrate in person. Unlike Kirschenbaum's artificial alloys, the audience is witness to Zuckerberg's 'sweat of the brow', but the hard mechanics of coding – the object code as it is run – are not imaged, or imagined. The action of the computational appears

metaphorically in the form of the immediate action of the Harvard 'network' as the software goes viral. In addition, the scene is one of carefully framed anticipation: Zuckerberg is shown writing the script, the source, but what the audience knows to be the object – the reordering of networking through social media – can only be known in retrospect; it is, as of the film's 'present', unknowable.

The structure of this scene offers a reflexive gesture at its own conceit. The voice-over explains the procedures of code, framing the visually disconcerting switches between scenes, and screens, with a smooth technical narrative. Here the dialogue that refers to code is for the most part technically authentic – the only changes Mark Zuckerberg requested to be made to the script were those referring to the programs and algorithms he used to build the initial site.⁶ The voice-over has the added effect of distinguishing the two writing procedures – giving human inflection to the blog post, which in turn functions as a way for the viewer to understand what is happening in the representation of code writing. The code is only named and therefore technically understood through the blog: 'definitely necessary to break out the Emacs and modify that Perl script'. Its appearance on screen is not legible – it is depicted as blurry lines of indistinguishable characters. In contrast, the blog text as seen on screen is perfectly legible – the audience can read along with the voice-over. In this section of *The Social Network*, the blog functions as the narrative frame for that which resists narrative, code. This happens visually (the viewer can read the blog but not the code script) and verbally (the viewer does not necessarily know what 'Emacs' or 'Perl' refer to but they can follow the context).

Both the code and the blog are woven into a nexus of writing, and reading, as latent. That is to say, although the cutting between programming and party produces an impression of immediacy, it also points towards the programming of Facebook as something of potential that has yet to be executed, as something that will become something else entirely when it runs. The foundations are laid in this scene for the future website Facebook, and for the emotional trajectory of the protagonist in the film's narrative arc. And so this scene, although conforming to a medial ideology of tech-geeks and code-writing, is also fraught with the anxiety of distance between the user and code. This anxiety is represented textually – through multiple representations of reading, writing and narrative. The counterpoint to these multiple depictions of reading and writing is one that epitomizes the difficulty in narrativizing code. And yet even with such an accessible, cinematic frame, this representation does not technically render the code readable or understandable: it merely conflates the difference between seeing and reading, or viewing and understanding.

<H1>Code as film (*Catfish*)</H1>

Catfish is a documentary film about New York photographer 'Nev' Schulman and a family with whom he develops a relationship via Facebook, after the daughter 'Abby' sends him a painting she has made of one of his photographs. The film is directed, filmed and produced by Ariel Schulman and Henry Joost, who began documenting Nev's life around the time he first received communication from Abby. *Catfish* is not explicitly about the material properties and forms of digital media but it is entirely about how digital media mediate human engagement. Whilst the film does not attempt to

depict code or the act of writing code (as is the case in *The Social Network*), it does task itself with somehow representing the abstract nature of new media relations through a human context. It does this by occupying the viewpoint of the computer user. Most of the narrative is conveyed through images of screens – desktops, laptops, phones. The film implicitly asserts that any performance of 'real life' must include the ways that real life is mediated, by filming the monitors that are the interfaces of our digital encounters. In addition, as a film that has been, for the most part, shot on portable digital cameras, it brings to the surface its own materiality as digital media. This draws the audience's attention to contemporary cinema as digital cinema, as computation and code, in a way that is suppressed, or at least elided, by the formal structure of *The Social Network*.

The main storyline of *Catfish* is plotted around the generic conventions of a thriller. As the story progresses, the audience may be dimly aware of an alternative narrative concern: repeated shots of screens on screens manifest a visual attention to surfaces and boundaries, borders and mediations. The numerous shots of screens within the frame of the screen the film is viewed on serve to emphasize the way the human subjects of the film are delineated by their position in front of a computer screen. *Catfish* represents digital technology almost in direct opposition to the medial ideology outlined by Kirschenbaum. Gone are the imagined cyberspaces of *Neuromancer* and *TRON* – the hidden depths of the digital realm. In their place is the boundary of the screen. The camera in *Catfish* continually zooms in on the various screens that represent communication between the subjects: but this zooming in can only make

more visible the surface; the camera cannot take the audience past the pixels that comprise the image on screen.

To this end, rather than attempt to humanize communication that is digitally mediated, or speculate about an imagined digital mechanics, *Catfish* fetishizes the modes of visual representation that computer coding produces. It pays attention to the graphics that make up the surface (the point of human encounter) of digital media. The film is full of images of pointers scrolling and selecting; shots of text as it is typed, and as it appears in message boxes; screens within screens; and photoshopped pictures. There is an inversion of a science-fictionalized representation of technology. Instead of immersion, *Catfish* posits the graphical language of digital technology as a wall between the viewer and technology, a barrier to their understanding and access to the 'truth'. The network that the computer substantiates (the Internet) and the network that it symbolizes (the friends of Facebook) are pointedly obscured by the screen. There are repeated images of screens as pixels – the camera zooms in on a screen but it cannot get behind the screen – the image breaks down into tricks of light. These scenes point to digital communication as mediation. In *Catfish*, code is not the subject, but the artificiality of digital communication is. Arguably, this artificiality is in fact a representation of the inaccessible world of computers – the non-readability of code. *Catfish* makes the visual surface of digital communications the visual surface of the film, sidestepping the impetus to position the technology itself as narrative or within narrative – and drawing attention to its own coded materiality.

The moments in the film when this structure is the most prevalent are the driving scenes. In the second half of the film, Nev goes to visit Abby. Nev, Ariel and Henry fly from New York City to Vail, Colorado, for a filming job, and then decide to drive a detour to Ishpeming, Michigan, to pay a surprise visit to Abby. By this point in the film, the seeds of doubt as to the authenticity of the Abby figure have been sown for the audience, and so the film moves along the lines of a generic road-trip thriller. The scenes of the boys' road trip are composite images combining 'real-life' footage from the car and graphics from Google Maps and Google Street View. For a piece on *Catfish* for *Time.com*, Mary Pols (2010: n.p.) writes of this sequence:

The trio is guided by GPS and Google Maps and spurred on by images of places they recognize from Facebook; their journey is illustrated with jaunty blue arrows that yank us around Michigan. (You'd be hard-pressed to find a film that more closely approximates our computer-dependent existence.)

The effect of these composite images is disconcerting. It pulls the viewer away from the drama of the present moment and towards the overarching theme, that digital technology as a medium can act as a barrier to the real. Rather than a sense of travel and movement, of perspective that pulls the viewer's eyes to the horizon, the audience is shown a flattened map – not a landscape. In a sense, the imposition of Google applications at this point in the film perhaps draws more attention to the constructs of cinema than it does to the workings of digital media. More particularly, the way the

Google applications disrupt the effect of a smooth, integrated composite image may be more a disruption of expectations about digital film than it is an intervention in a cultural appreciation of digital aesthetics. As Sean Cubitt (2004: 146) has noted:

... audiences have a clear idea in their minds when offered digital entertainment: a certain seamlessness, a generic expectation of something new, a willingness to sever connections with fundamental laws of nature. It is not necessarily the essential nature of digital media to provide these things, but they often are provided and become part of the cycle of expectation and satisfaction on which the closed loop of commodity production proceeds.

In *Catfish* this cycle is clearly upended: it is a film about software told through the documentation of people as they go about their day-to-day lives (on film). In scenes such as the road trip, this is made clear: the digital is shown to augment the 'reality' being documented. This draws attention not only to the instability of the narrative as documentary, but moreover to the filmic process as smoothly integrating various digital processes. This perhaps increases the audience awareness of the code that is being depicted as agent (in the story) and as running the story (as the film). The layering of graphic animation draws attention to the images' composite structures. As much as in these scenes the representation of Google Maps might lead us to rethink cinematic representations of the road (this is perhaps an important intervention in the genre of the 'road movie' – explicitly representing the banality of always being *on* road, and *on* grid

through GPS), it also reminds us of the visual codes used to produce Google Maps, of the abstract nature of such an application. Although *Catfish* does not explicitly deal with code, it offers, and is itself, an important example of the way cinema might play with cultural representations of the digital, and a useful way into thinking about how these representations function in cinematic narrative.

Whilst Kirschenbaum speaks derisively of 'medial ideology', the intention of this article has been to suggest that a medial ideology is perhaps our only way of representing what is beyond the screen. What *Catfish* does is to posit the screen itself – and the pixelated graphics that comprise the scene of the screen – as a sticking point for human encounters with the computer. In this way, *Catfish* seems to point its viewer towards precisely that which cannot be seen: not only the 'real' identity of the Facebook user, but, moreover, to invert van den Boomen's phrase, it renders relevant 'what remains invisible', and renders thinkable 'what remains unmetamorphised'. The film highlights the complexity of attempting to visually represent something deeper about a technology that is predominantly experienced as a composite surface.

Conclusion: Code in the popular imagination

Where *Catfish* shows us the screen, *The Social Network* constructs a complicated montage. Both seem to represent for the audience the digital medium as material. They zoom in on the physical thing, or elucidate untranslatable processes. But they also both highlight the way the material facts of digital technology can slip away from the surface the minute we begin to draw them there. Representations of cyberspace in science fiction and cyberpunk traditions are undoubtedly an imaginary misnomer. But they do

not purport to be anything else. Likewise in *Catfish* and *The Social Network*, there is no direct attempt to represent code, or coding processes beyond the screen. This might be an aesthetic choice, but that in itself is telling. As Zadie Smith (2010: n.p.) suggests: 'even if we spent half the film looking at those busy screens (and we do get glimpses), most of us would be none the wiser'. Code as a fact of computer mechanics can only ever be represented through abstraction; it resists easy assimilation with narrative, and demands complex cultural representations.

In popular imagination, the production of code by humans happens in the same configuration as the production of any digital text by humans (as *The Social Network* demonstrates). Rather than critique these representations as themselves misleading, it could be said that they attempt to reflect what is misleading in our everyday encounters with code. This is to suggest that our own encounters with the digital do not demand we pay attention to mechanics and 'sweat of the brow'; instead, they are predicated on the ease with which we suppress awareness of actual digital processes. Arguably, the screen as barrier and disguise in *Catfish* is a visual representation of this assumption. In the book *Behind the Blip*, Matthew Fuller considers a similar conceit in relation to the process of writing with a computer. Fuller (2003: 146) argues that 'the two forms of language [word processing and code] are becoming increasingly close. The most obvious similarity is that before being compiled, code is written text, characters in a row ...' This assertion of Fuller describes the kind of conflation performed in the programming sequence of *The Social Network*.

In the article 'Digital Ontologies', Johanna Drucker productively describes the theoretical concerns invested in recent discussions of code and representation. Drucker (2001: 144–45) suggests:

What is at stake is not the question of whether there is a 'truth' to this idea that the stored 'code' exists and can be made use of without graphic manifestation, and that it is stored materially. What is at stake is that this idea pushes the cultural status of the digital to a place ... in which the sense of an inevitable and seamless interchangeability replaces the idea of differentiated and resistant material instantiation of form.

Despite acknowledgement that the independent fact of code persists as a problematic, we are continually drawn to attempting to critique and represent the ways in which code is 'a resistant material instantiation of form'. Drucker and Kirschenbaum are aligned in their call for more evolved readings of new media culture – readings that can incorporate the nascent textual, mechanical and non-visible properties of code. This article has suggested that careful representations of code can elucidate, rather than obscure, the oscillating functions of image and text that mark critical approaches to code. Considering the representation of code in the films *The Social Network* and *Catfish*, through the framework of a materialist approach to the digital, offers an insight into how these stakes play out in popular culture.

Notes

References

- Bakardjieva, M. (2005), *Internet Society: The Internet in Everyday Life*, London: Sage.
- Bassett, C. (2007), *The Arc and the Machine: Narrative and New Media*, Manchester: Manchester University Press.
- Berry, D. M. (2008), 'A contribution towards a grammar of Code', *The Fibreculture Journal*, 13, n.p.
- Bolter, J. D. (2003), 'Critical theory and the challenge of new media', in M. E. Hocks and M. R. Kendrick (eds), *Eloquent Images: Word and Image in the Age of New Media*, Cambridge, MA: MIT Press.
- Cayley, J. (2002, September), 'The code is not the text (unless it is the text)', *Electronic Book Review*, n.p.,
<http://www.electronicbookreview.com/thread/electropoetics/literal> (accessed 18 February, 2011).
- Chun, W. H. K. (2005), 'On software, or the persistence of visual knowledge', *Grey Room*, 18, pp. 26–51.
- . (2008), 'On "sourcery," or code as fetish', *Configurations*, 16: 3, pp. 299–324.
- . (2011a), *Programmed Visions: Software and Memory*, Cambridge, MA: MIT Press.
- . (2011b), 'Crisis, crisis, crisis, or sovereignty and networks', *Theory Culture Society*, 28: 6, pp. 91–112.

'Code, n.1.' (March 2011), OED online. Oxford University Press.

<http://www.oed.com/view/Entry/35578?rskey=iS7Ojy&result=1&isAdvanced=false>
(accessed 26 April, 2011)

Cubitt, S. (2004), *The Cinema Effect*, Cambridge, MA: MIT Press.

Drucker, J. (2001), 'Digital ontologies: The ideality of form in/and code storage: Or: Can graphesis challenge mathesis?', *Leonardo*, 34: 2, pp. 141–45.

———. (2009), 'A review of Matthew Kirschenbaum, mechanisms: New media and the forensic imagination, Cambridge, MA and London, UK: MIT University Press, 2008', *DHQ: Digital Humanities Quarterly*, 3: 2, n.p.,
<http://www.digitalhumanities.org/dhq/vol/3/2/000048/000048.html>, (accessed 22 May, 2013)

Fincher, D. (2010), *The Social Network*, Columbia; USA.

Fuller, M. (2003), *Behind the Blip: Essays on the Culture of Software*, New York: Autonomedia.

Galloway, A. (2004), *Protocol: How Control Exists after Decentralization*, Cambridge, MA: MIT Press.

———. (2006), 'Language wants to be overlooked: On software and ideology', *Journal of Visual Culture*, 5: 3 pp. 315–31.

———. (2011), 'Are some things unrepresentable?', *Theory Culture Society*, 28: 7–8, pp. 85–102.

Gibson, W. (1984), *Neuromancer*, London: Harper-Voyager.

Hayles, K. (1999), *How We Became Posthuman: Virtual Bodies in Cybernetics, Literature and Informatics*, Chicago: University of Chicago Press.

———. (2006), 'Traumas of code', *Critical Inquiry*, 33: 1, pp. 136–57.

Haythornwaite, C. and Wellman, B. (eds) (2002), *The Internet in Everyday Life*, Oxford: Blackwell Publishing.

Joost, H. and Schulman, A. (2010), *Catfish*, Relativity Media, Rogue Pictures, Universal Pictures; USA.

Kirschenbaum, M. (2003), 'The word as image in an age of digital reproduction', in M. E. Hocks and M. R. Kendrick (eds), *Eloquent Images: Word and Image in the Age of New Media*, Cambridge, MA: MIT Press, pp. 137–58.

———. (2008), *Mechanisms: New Media and the Forensic Imagination*, Cambridge, MA: MIT Press.

Kitchin, R. and Dodge, M. (2011), *Code/Space: Software and Everyday Life*, Cambridge, MA: MIT Press.

Kittler, F. (2008), 'Code (or, how you can write something differently)', in M. Fuller (ed.), *Software Studies/A Lexicon*, Cambridge, MA: MIT Press, pp. 40–47.

Laurel, B. (1991), *The Art of Human–Computer Interface Design*, Menlo Park, CA: Addison-Wesley.

———. (1995), *Computers as Theatre*, Menlo Park, CA: Addison-Wesley.

Mackenzie, A. (2006), *Cutting Code: Software and Sociality*, New York: Peter Lang.

Mackenzie, A. and Vurdubakis, T. (2011), 'Codes and codings in crisis: Signification, performativity and excess', *Theory Culture Society*, 28: 6, pp. 3–23.

Final article published as: Dinnen, Z. (2013), 'Breaking out that Perl script: The imaging and imagining of code in *The Social Network* and *Catfish*', *European Journal of American Culture*, 32: 2, pp. 173-186, doi: 10.1386/ejac.32.2.173_1

Negroponte, N. (1995), *Being Digital*, London: Hodder and Stoughton.

Oshii, M. (1995), *Ghost in the Shell*, Manga Entertainment; UK.

Pols, M. (2010), 'Fish tale',

<http://www.time.com/time/magazine/article/0,9171,2019606,00.html>. Accessed

15 April 2011.

Smith, Z. (2010), 'Generation why?', *The New York Review of Books*, 57: 18, n.p.

Sorkin, A. (2010), 'A look at *The Social Network*', Interview by Charlie Rose, *Charlie Rose*, New York: Bloomberg Television.

Turkle, S. (1997), *Life on the Screen: Identity in the Age of the Internet*, New York: Touchstone.

van den Boomen, M. (2009), 'Interfacing by material metaphors: How your mailbox may fool you', in M. van den Boomen et al. (eds), *Digital Material: Tracing New Media in Everyday Life and Technology*, Amsterdam: Amsterdam University Press, pp. 7–20.

Wachowski, A. and Wachowski, L. (1999), *The Matrix*, Warner Bros. Pictures; USA.

Wiseman, L. (2007), *Die Hard 4.0*, 20th Century Fox; USA.

<H1>Suggested citation</H1>

¹ The distinction between source code and object code is articulated most clearly by Wendy Hui Kyong Chun in her work on code and software (see Chu 2008, 2011). I will be referring to Chun's work in detail later in the article.

² Positing the screen and more particularly the GUI as the point of human–computer interaction (HCI) can be traced back to early writing on digital culture by Nicholas Negroponte and Brenda Laurel (see Laurel 1990, 1991; Negroponte 1995).

³ For Chun, the analysis of source code and object code as distinct serves to enable a fuller critique of digital culture: first, to consider the way in which software embeds particular cultural assumptions within technological use and practice; and second, to consider the way in which the human programmer is figured and identified in the technology industry and through culture more broadly (see Chun 2011a, 2004).

⁴ A literary equivalent of these examples might be Neal Stephenson's fiction, in particular *Snow Crash* (1992) and *Cryptonomicon* (1999). Both of these texts connect the technical and imaginative in the context of the everyday – albeit an everyday, like some of the films listed, represented through the estranging generic conventions of science fiction (a particular framework this article is not interested in reproducing) – and are ripe perhaps for the kind of reading Kirschenbaum advocates. Here, though, my focus is on the visual and cinematic, and how these particular kinds of narratives highlight the tension between code as performative, code as graphic sign and code as text.

⁵ The use of 'anthropomorphize' in this context is taken from Chun's reading of Galloway's writing on software and code. In 'On Sourcery', Chun suggests that the title of Galloway's 2006 article, 'Language *Wants* to Be Overlooked', undermines his attempts to figure code outside of a linguistic framework: 'The inevitability of this anthropomorphization is arguably evident in the title of Galloway's article: "Language *Wants* to Be Overlooked" ' (Chun 2008: 305, emphasis added).

⁶ Aaron Sorkin made this point in an interview with Charlie Rose, September 2010 (cited in references).